applied sciences

MDPI

Article

Digital Simulation and Identification of Faults with Neural Network Reasoners in Brushed Actuators Employed in an E-Brake System

Gouri Ramesh [1,*], Pablo Garza [1] and Suresh Perinpanayagam [2]

[1] School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield MK43 0AL, UK; pablo.garza@cranfield.ac.uk
[2] Integrated Vehicle Health Management Centre, Cranfield University, Cranfield MK43 0AL, UK; suresh.nayagam@cranfield.ac.uk
* Correspondence: g.ramesh@cranfield.ac.uk

Abstract: The aerospace industry is constantly looking to adopt new technologies to increase the performance of the machines and procedures they employ. In recent years, the industry has tried to introduce more electric aircraft and integrated vehicle health management technologies to achieve various benefits, such as weight reduction, lower fuel consumption, and a decrease in unexpected failures. In this experiment, data obtained from the simulation model of an electric braking system employing a brushed DC motor is used to determine its health. More specifically, the data are used to identify faults, namely open circuit fault, intermittent open circuit, and jamming. The variation of characteristic parameters during normal working conditions and when faults are encountered are analysed qualitatively. The analysis is used to select the features that are ideal to be fed into the reasoner. The selected features are braking force, wheel slip, motor temperature, and motor angular displacement, as these parameters have very distinct profiles upon injection of each of the faults. Due to the availability of clean data, a data-driven approach is adopted for the development of the reasoner. In this work, a Long Short-Term Memory Neural Network time series classifier is proposed for the identification of faults. The performance of this classifier is then compared with two others—K Nearest Neighbour time series and Time Series Forest classifiers. The comparison of the reasoners is then carried out in terms of accuracy, precision, recall and F1-score.

Keywords: IVHM; fault detection; diagnostic reasoner; neural networks; digital twin; landing gear; aircraft

Citation: Ramesh, G.; Garza, P.; Perinpanayagam, S. Digital Simulation and Identification of Faults with Neural Network Reasoners in Brushed Actuators Employed in an E-Brake System. *Appl. Sci.* **2021**, *11*, 9171. https://doi.org/10.3390/app11199171

Academic Editor: Dario Richiedei

Received: 12 July 2021
Accepted: 20 September 2021
Published: 2 October 2021

1. Introduction

Ever since the inception of commercial aviation, civil aircraft have been relying on gas turbine engines to power up their systems. As an industry that survives on wafer-thin profits and is bound by scores of safety regulations, all stakeholders, from manufacturers to operators, are continually looking to improve their methods to extract maximum efficiency from both the machines and humans they employ. In line with this goal, they are constantly testing new technologies to either replace existing systems or integrate with them. Two technologies that are rapidly and extensively being tested and deployed for better performance are more electric aircraft (MEA) and integrated vehicle health management (IVHM), as well as digital twins for different levels of fidelity in a variety of applications ranging from system development to vehicle health monitoring [1–4].

Grounding the aircraft for activities pertaining to maintenance make it a non-performing asset and causes the aircraft operators to lose out on considerable revenue. However, owing to its paramount importance, maintenance work in the aerospace industry is undertaken at regular intervals according to pre-determined schedules. This means that every aircraft, irrespective of its current health or its propensity to develop a fault, is grounded according

to schedule. Although discovering unexpected faults during regular maintenance will prevent dangerous situations from happening, this finding will most likely lead to an extensive delay in the maintenance schedule. This is because the resources to resolve the fault may not be immediately available. This, in turn, can cause a potential disruption in the maintenance schedule and overall airline operation, affecting the overall costs of the operation [5].

Monitoring an aerospace vehicle's health can help evaluate the condition of the systems, prevent unnecessary maintenance, and increase the vehicle's useful life. In case abnormal equipment behaviour is observed during the vehicle operation, proper arrangements can be made to address the issue and decrease the probability of a dangerous situation. By using the data generated by the health monitoring systems, original equipment manufacturers (OEMs) benefit by using the data generated by the aircraft to improve the existing systems for more optimal performance or develop new ones more suited for operation [6].

With the increasing availability of data, development of technology with the internet-of-things (IoT) components, and the realisation that such data can be used to make better decisions for system design and maintainability, the field of IVHM is attracting high-profile stakeholders who are keen to determine and tap its full potential. Prognostics and diagnosis of aircraft systems carried out by health monitoring assets provide such stakeholders with valuable data to detect abnormalities, optimise maintenance schedules, and for better estimations of the system remaining useful life (RUL) [6–8].

For fault identification, reasoners developed from neural network (NN) algorithms become an asset for health management applications. By adequately training such reasoners, data obtained from the sensor systems can be used accordingly to provide insight into the overall system performance and modify the maintenance regime. A reasoner with good accuracy helps achieve the dual objective of saving time as well as other resources, such as manpower since it could potentially eliminate manual inspections of the target system [1,9,10].

The objective of this paper is to develop and propose a reasoner with good accuracy to identify selected faults that can occur in an aircraft electric braking system (EBS) developed from three different machine learning (ML) algorithms. The introduction of electric actuators on MEA platforms introduces new fault modes to systems that require further study to cement its applications on modern aircraft. It studies the characteristics of the parameters provided by an EBS digital model working under ideal conditions and induced fault modes. This is followed by the identification of appropriate time series attributes to eliminate redundancies from being fed to the data-driven reasoner. Finally, the comparison of the algorithm's performance is undertaken for further development for an EBS reasoner.

2. Literature Review

Landing gear or the undercarriage of an aircraft is one of the critical systems in an aircraft, especially during taxiing, take-off and landing. It performs vital functions, such as supporting the weight of the aircraft, absorbing impact upon touchdown, and providing braking and directional control. Typical parts of a landing gear include oleo strut, tyres, steering actuator, up and down locks, trailing arm or telescopic legs, and retracting actuator [11]. The braking system provides the braking action, which reduces braking distance and consequently increases payload capacity [12].

Commercial aircraft have primarily relied on hydraulic and pneumatic systems for most of their actuation systems. Recent commercial aircraft platforms have seen the introduction of electromechanical actuation systems in an attempt to utilise electrical power in more systems due to potential favourable advantages, such as a reduction in weight, fuel costs, and operating costs. Moreover, electrical power can be stepped up and down, stored, and, in turn, controlled and distributed to other systems easily according to their requirements due to advancements in the field of power electronics [4]. With the introduction of a 100 per cent electrical actuator in the A380's thrust reversal system

and B787 replacing its pneumatic circuit with an electrical counterpart in the braking system propelled, the move towards More Electric Aircraft, thereby making the possibility of achieving All-Electric Aircraft, appears plausible in the near future [2]. Other major players in the industry have also adopted the MEA approach, with SAFRAN developing a fully electrical braking system and BAE introducing electrical flight controls to military aircraft [12,13].

The incorporation of more electric components and actuators demands greater electric power generation, thereby requiring cooling and control systems of greater complexity. It is also worth noting that a significant weight reduction benefit of switching from hydraulic/pneumatic to electric actuation will only be realised once significant improvements have been made to the current power electronic modules. Another major challenge with MEA is evaluating the safety and reliability of the new designs. The cost and effort required to secure the initial certification for these new designs are considered prohibitively high, as the aerospace industry relies very much on legacy certification. Hence, the use of digital technology for system design and testing acquires a bigger dimension for new technology introduction [14–16].

Integrated Vehicle Health Management (IVHM) was first proposed by NASA in 1992 to determine the health of systems before, during and after their period of operation, with the goal being to avoid catastrophic failures. By introducing IVHM, NASA hoped to reduce maintenance and labour costs, and increase reliability and availability of equipment [17]. IVHM helps evaluate the health of systems in the aerospace sector through four approaches: Diagnosis, Prognosis, Automated Inspections, and Anomaly Detection.

IVHM aims to provide a holistic approach to vehicle health management. Until its introduction, the health of individual components was monitored. The goal of IVHM is to assess the overall health of the vehicle by integrating the health status of various components and taking the inter-dependability of systems into account. IVHM intends to achieve this by incorporating various hardware and software into the system, such as sensors, reasoning algorithms, maintenance planners, databases for storing health parameters, etc., which aid in monitoring the health during operations.

Fault diagnosis aims to identify the fault that has occurred from the control characteristics and sensor data available. It involves monitoring system characteristics for anomalies and analysing them to determine the type and nature of fault that has occurred. Fault prognosis focuses on predicting the future health condition of the system. Prognosis is mainly used to calculate RUL and predict any faults that are likely to occur in the future. Various reasoning methods are used for the purpose of fault diagnosis and prognosis and the same are discussed by T. V. Tung and B. Yang [18]. Reasoners or reasoning systems are among the many tools that may be utilised to carry out a diagnosis. They are algorithms that use various strategies known as reasoning strategies to deduce faults. C. M. Ezhilarasu et al. [1] introduces various reasoning strategies and systems for fault detection in her paper.

3. Modelling and Simulation of an Aircraft Electric Braking System

The failure modes, effects and criticality of the various systems and components in an aircraft's landing gear with a brushed EMA are discussed in brief in this section. Considering how failure modes are assessed, a failure in the braking unit could be catastrophic as it could lead to the aircraft overshooting the runway and potentially the loss of lives from the incident. Similarly, a failure of the anti-skid braking system (ABS) would result in a significant reduction in braking performance, becoming a major critical event with potentially serious injuries and loss of life.

As a whole, the landing gear is a safety-critical system, and the braking unit is certainly one of the most important systems it hosts. In order to comply with MEA approaches, components such as electromechanical actuator (EMA) are suited to replace existing hydraulic braking systems and, with that, the necessity to study new failure modes attached to them. A number of possible failures of an EBS and their effects are listed in Table 1. IoT faults, more specifically to sensor/actuator and interface components faults,

are the only non-catastrophic events listed as they could potentially reduce the braking performance, but their role in feedback loops can cause the braking force to deviate from expected values.

Table 1. Simplified FMECA from an EBS.

Component	Failure	Failure Effects	Critical Assessment
Electric Motor (EMA)	Winding Damage Jamming Overload	Overheating Loss of Torque	Catastrophic
Electrical Wiring	Open Circuit Short Circuit Intermittent Open/Short Circuit	Partial/Total Loss of Power Thermal Runaway	Catastrophic
Leadscrew (EMA)	Worn-Out Threading Structural Damage	Control Loss of the EMA No Force Applied on Brake	Catastrophic
IoT	Calibration Faults Bias Fault	Incorrect Sensor Readings	Major

3.1. Electric Braking System Simulation

For developing an appropriate reasoner for the EBS, a substantial amount of data is required in order to train the system to detect any anomalies. An EBS model was provided by [19], developed in the MATLAB/Simulink environment with Simscape components. The model employs a brushless DC motor to actuate the EMA in the aircraft's single EBS with ABS included serving the purpose of providing relevant data for training the reasoner. The environment in which the simulation is working is on ideal conditions, i.e., no external or environmental conditions are affecting the braking. The model is a simple representation of a single electromechanical actuator providing the necessary braking force to an aircraft. An ABS system is included in the model to give the simulation a more realistic approach. The model used has three main layers, starting from the overall aircraft speed calculation layer, the ABS controller, and the EMA layer. An overview of the model is found in Figure 1.

A DC brushed motor block from the Simscape environment provides electric and torque parameters useful for the model. Usually, a four-actuator brake per wheel would be used in real applications, increasing the system redundancy and robustness. The failure of one actuator on a four-EMA system will certainly reduce the effects on aircraft braking, but the purpose of having this one-EMA model is to obtain more visual and impactful information for data analysis [20].

A number of sensors are integrated into the model to verify that the components are performing their intended functions, evaluate the EBS performance as a whole, and detect abnormal conditions. Temperature, torque and force sensors are included in the model and are the main source of data for the reasoner. Starting with temperature, an electric motor overheating may occur due to a number of causes mentioned previously, such as jamming, bearing and winding failures, over-voltage or other poor environmental conditions [21].

A force sensor is integrated into the model to evaluate the actuator performance and the force applied to the brake. Torque sensors are also located in the leadscrew, and it is used to gauge the torque provided by the EMA electric motor. Other specific resultant data obtained from the model are the brake pressure, resultant torque (difference between wheel torque and brake torque), aircraft and wheel velocity, wheel slip and stopping distance. Table 2 shows a list of signals and data obtained from the model.

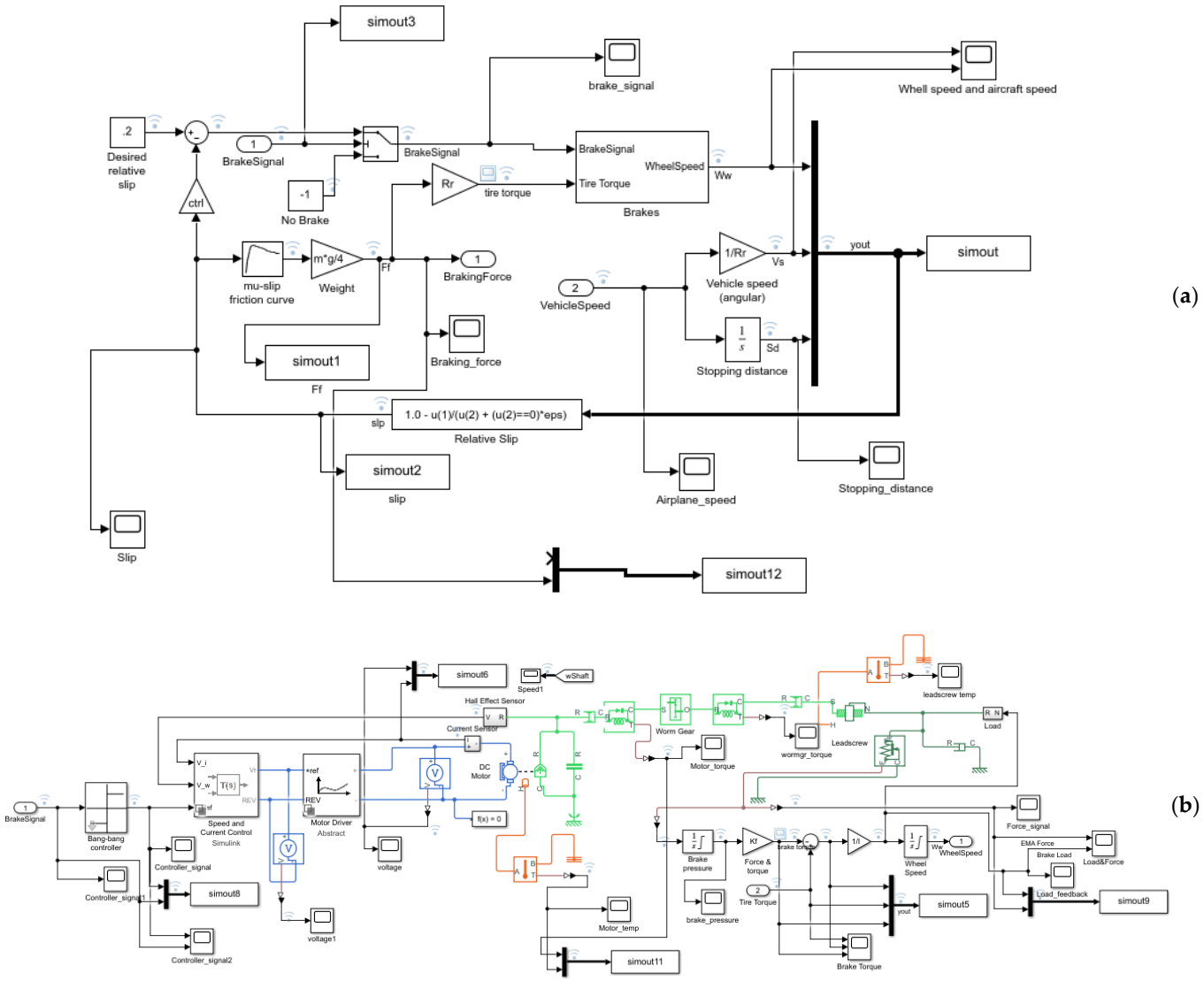

Figure 1. Overview of the EBS Model (**a**) ABS Control Layer; (**b**) EMA and Actuator Controller Layer.

Table 2. Data Obtained from EBS Model.

Data
Braking Force
Wheel Slip
Electric Motor Temperature
Motor Shaft Angular Displacement
Wheel Angular Speed
Aircraft Speed
Stopping Distance
Brake Signal
Resultant Torque
Motor Power
Brake Torque
Tire Torque
Shaft Speed

The model uses generic values for the simulation to generate useful data for the reasoner. Some parameters used include the initial aircraft linear velocity of 88 m/s, landing mass of 8000 kg, wheel radius of 0.34 m, and maximum brake pressure of 30,000 Pa.

The simulation runs for 60 s, with the braking procedure starting 1 s after the simulation starts. Results from the model aircraft behaviour are shown in Figure 2.

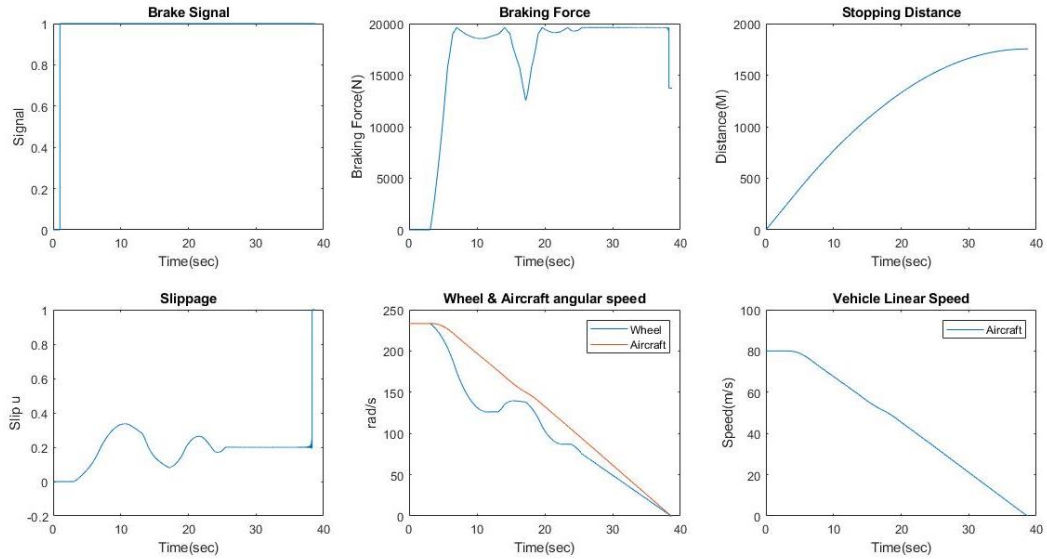

Figure 2. Aircraft Braking Data.

Once the braking signal is set, the aircraft speed decreases smoothly with time, showing wheel slip being stabilised by the ABS and the vehicle coming to a halt at around 1800 metres. Figure 3 shows more specific data obtained from the EMA, such as motor power, torque, force and shaft displacement. It is noticeable how the ABS and motor controller fluctuate to obtain a smooth braking procedure.

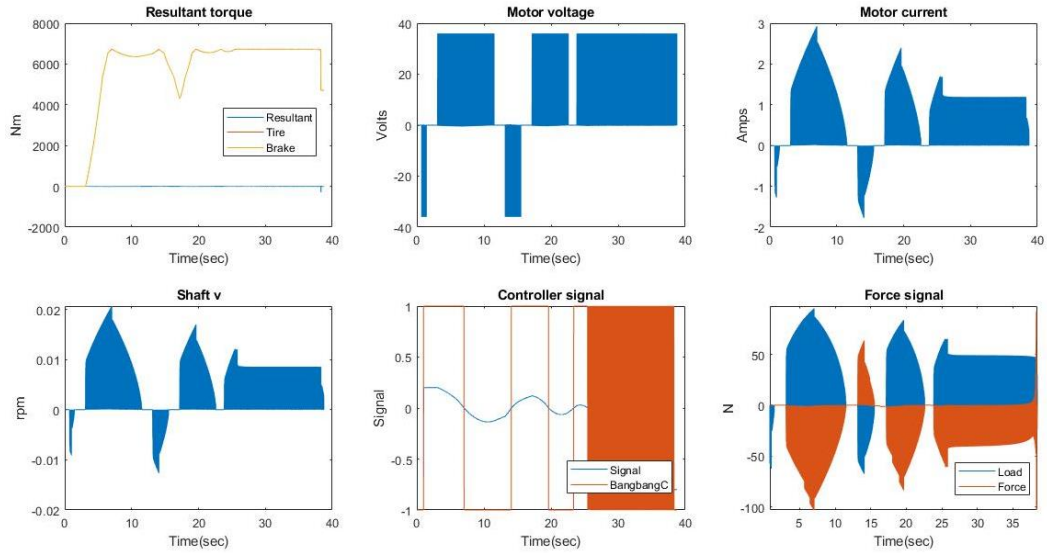

Figure 3. Actuator and ABS Data.

3.2. Fault Mode Integration and Simulation

Having obtained the EBS normal performance model data, fault modes are selected according to their criticality and the variety of effects they could provide for data analysis purposes. Electrical and mechanical faults being critical, an electric motor open circuit (OC), intermittent open circuit (IOC) and motor jamming faults were selected for data processing. Undesirable situations as a result of these fault modes could lead to catastrophic accidents.

Thus, the study of these faults is important to assure IVHM capabilities and increase the systems' reliability.

The jamming condition is simulated by applying increased friction in the motor shaft. Some noticeable results are the lack of braking force as the actuator does not make contact with the brake, disturbance in the wheel slip, and no change in the aircraft velocity. Motor power is greatly increased as the actuator controller battles to overcome the jamming condition. This situation certainly has the potential to turn into a catastrophic accident.

The OC and IOC are integrated into the model by opening the circuit with a switch. The OC opens 20 s after the braking procedure starts, halting the actuator activity completely, and displaying a complete disruption in the aircraft braking and the actuator power. Similarly, the IOC is set to occur every 5 s, which halts the EBS activity intermittently but still managing to stop the aircraft with a noticeable reduction in performance. Some data comparing the OC and IOC are shown in Figure 4.

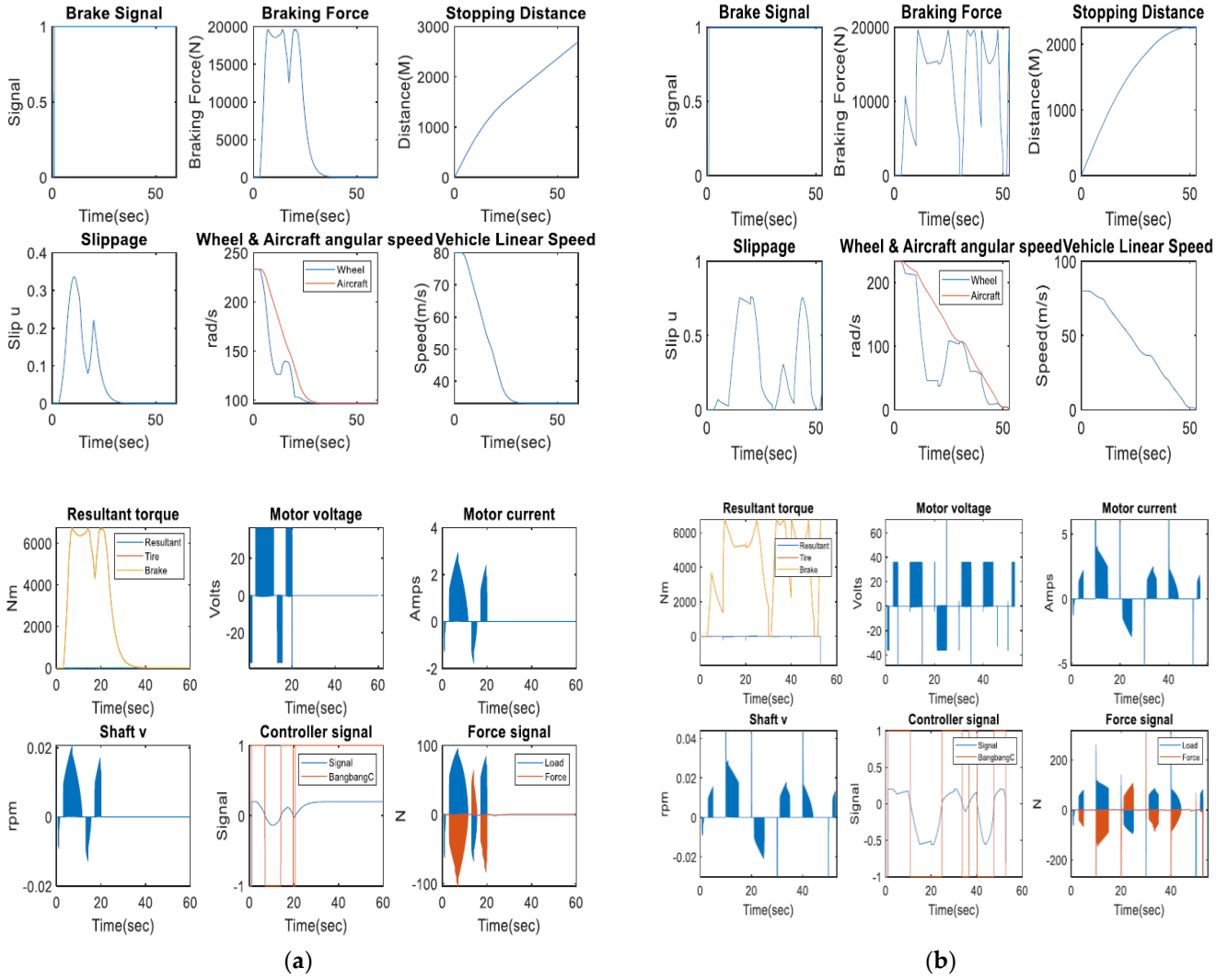

Figure 4. Comparison in Resultant Data from OC and IOC Faults: (**a**) OC Fault Resultant Data; (**b**) IOC Fault Resultant Data.

For the purpose of the development of an accurate reasoner to diagnose faults, the parameters that have unique characteristics and best represent a fault are of particular interest. It is desirable to obtain data and evaluate redundancies in order to reduce the computational load without compromising on accuracy.

4. Reasoner Development

Using the seven-step process presented by [22], ML models are used for the reasoner development. The process starts with data gathering, preparation and model selection. It continues with model training and evaluation and finishes with parameter tuning and prediction. The performance of a classifier is often measured by its accuracy, but the reasoner performance can also be measured by precision rate, recall rate and F1 score, and their formulas can be found in Equations (1)–(4):

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total No. of Predictions}} \tag{1}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{2}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{3}$$

$$\text{F1} - \text{Score} = 2\,\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

After evaluation, the parameters related to the model are fine-tuned. Necessary changes are made to them, which could potentially increase the model performance. In real-world applications, supervised learning is commonly employed for fault detection and diagnosis, where the input and expected output are available. The process of fault isolation and diagnosis is fundamentally a pattern recognition problem. When signatures of a fault are used to map it to an exact fault case, the process becomes a classification problem [23].

The performance of a classifier is often represented in the form of a confusion matrix. It is often referred to as an error matrix or confusion chart and represents the actual labels of instances in the dataset versus the label assigned to it. Figure 5 shows the format used for the EBS reasoner development. The table shows the number of true positives, true negatives, false positives and false negatives generated by the classifier. The diagonal elements in the confusion matrix indicate the correct predictions made by the classifier. The entire process of reasoner development is illustrated in Appendix A.

Predicted Class / True Class	A	B	C
A			
B			
C			

Precision	1 – (Precision)

Recall			
1- Recall			

Figure 5. Confusion Matrix for Multiclass.

4.1. Data Generation and Feature Selection

Data were extracted such that the faults occurred at various instances of time in the process of braking. This means that the velocity of the aircraft at the time of occurrence of fault varies throughout the dataset. The data provided are in the form of a time series. Up to nineteen such possible input parameters are available from the simulation of the model. The time interval between data points generated is 0.5 s, and the total number of data samples used in this case is 120. The mode of the length of the data series is 121, and the available data are split into training and testing datasets in a 3:1 ratio. The split is random, and care was taken to ensure that the test and train datasets did not contain the same cases.

Efforts are made to include possible extreme case scenarios so that all possible cases within the distribution are addressed. Each series of data is classified into three depending on the condition they represent, as shown in Table 3.

Table 3. Data Obtained from EBS Model.

Feature Name	Label
EMA Electric Motor Open Circuit Fault	1
EMA Electric Motor Intermittent Open Circuit Fault	2
EMA Electric Motor Jamming	3

Features are quantified properties that are put into a model, and up to 19 different parameters are generated from the EBS model simulation, generating 19 factorial or 1.2×10^7 possible combinations as input features. Feeding all the features into the ML models are not a viable option due to the high number of combinations, which will translate into more processing time. In cases with a high number of data combinations, a trade-off between accuracy and processing time is considered.

The comparative study of the previous sections shows the braking force being different in the normal braking condition simulation and the three fault modes. The wheel slip profile shows major differences for each scenario and is a parameter derived from wheel and vehicle speed. The other parameters found with major variability are the motor angular displacement and motor temperature which tends to change at the earliest sign of an anomaly.

The braking force is used as the input feature for the univariate. For multivariate models, the number of features to be fed into the model was arbitrarily chosen as four. These four parameters are braking force, wheel slip, motor angular displacement, and motor temperature, as they display observable variation during each of the scenarios.

4.2. Long Short-Term Memory Reasoner

With the data from the EMA model simulation, the prospect of a reasoner employing Long Short-Term Memory (LSTM) is studied. The ability of remembrance demonstrated by this NN approach makes it of particular interest in applications related to forecasting and time series classification [24]. This ability comes from the incorporation of a memory cell in its architecture.. Each cell takes in an input, the previous cell state, the weight and biases parameters determine what values are passed on to the next cell and which data are retained or ultimately forgotten [25]. Formulas governing the LSTM model used can be found from Equations (5)–(10) [26]:

$$\text{Cell state, } \mathrm{c}_t = f_t \odot \mathrm{c} + i_t \odot g_t \tag{5}$$

$$\text{Hidden state, } \mathrm{h}_t = o_t \odot \sigma_c(\mathrm{c}_t) \tag{6}$$

$$\text{Input gate, } i_t = \sigma_g(W_i X_t + R_i h_{t-1} + b_i) \tag{7}$$

$$\text{Output gate, } o_t = \sigma_g\left(W_o X_t + R_g h_{t-1} + b_o\right) \tag{8}$$

$$\text{Forget gate, } f_t = \sigma_g\left(W_f X_t + R_f h_{t-1} + b_f\right) \tag{9}$$

$$\text{Cell candidate, } g_t = \sigma_c(W_o X_t + R_o h_{t-1} + b_o) \tag{10}$$

where W, X, R, h and b denote weight, input, recurrent weights, and biases. The gate activation function is represented by σ_g.

The use of LSTM is chosen for the experiment due to a number of reasons, including the ability to learn information in a considerably long time period, ability to remember previous states, LSTM's insensitivity to gap length, noise handling, and no need for fine-tuning of parameters [27,28].

MATLAB R2020b was used for the LSTM reasoner modelling. The implemented model consists of five layers which are namely the input, bi-directional, fully-connected, softmax and classification layers as shown in Figure 6. The input layer takes in the sequence followed by the bi-directional layer responsible for studying the dependencies through the length of the time series. The activation function for state and cell in this layer is a hyperbolic tangent function on which the sigmoid function dictates the gate activation function.

Figure 6. LSTM Layers Architecture.

The fully connected layer is a feed-forward neural network, and the number of hidden layers within this network is customisable. This is followed by a final layer of activation function, the softmax layer, which is a normalisation function that converts the output of the previous layer to a value in the range of 0 to 1. This layer then passes the values to the next classification layer, which calculates the cross-entropy between the prediction and the classified label.

Cross-entropy is defined as the extra bits required to represent an event from one distribution to another [29]. The lower the value of cross-entropy, the better the classification of the label. The cross-entropy function employed by the training network is defined in [30] and is shown in Equation (11).

$$Loss = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K} w_i t_{ni} \ln y_{ni} \tag{11}$$

in which N denotes the number of samples, i and w represent the class and weight, respectively. t_{ni} corresponds to the class of the nth sample and y_{ni} to the output for the nth sample of ith class. The learning rate optimiser employed in the model is an adaptive moment estimation. LSTM reasoner model parameters are shown in Table 4.

Table 4. LSTM Model Parameters.

Classifier	Type	Input Attributes	Training Set and Test Set
Univariate	Braking Force	Epoch: 50 Mini-batch Size: 3 Learning Rate: 0.001 Hidden Layers: 20 Solver: Adam	Training Set: 90 Instances (32 OC; 28 IOC; 30 Jamming) Test Set: 30 (10 Instances each per Fault)
Multivariate	Braking Force Wheel Slip Motor Temperature Motor Shaft Angular Displacement	Epoch: 50 Mini-batch Size: 2 Learning Rate: 0.001 Hidden Layers: 20 Solver: Adam	-

4.3. Time Series Forest Reasoner

The Time Series Forest (TSF) is an ensemble of time series trees (TST). Time series are split into random intervals, and features are extracted from each interval. Such features are the mean, standard deviation, and slope of the least square regression line. TSTs are constructed with these features as inputs. The splitting function used here, as proposed in [31], is called entrance gain, which is a measure of entropy and margin, described in Equations (12) and (13).

$$Entropy = -\sum_{c=1}^{C} \gamma_c \log \gamma_c \tag{12}$$

$$Margin = \min_{n=1,2,..,N} |f_k^n(t_1, t_2) - \tau| \quad (13)$$

where γ_c corresponds to an instance of cth class, f_k^n denotes nth instance of kth feature, t_1 and t_2 are starting and ending points of intervals, and τ is the threshold. The entrance gain is given by Equation (14). This TSF model consists of 100 trees. The multivariate model of TSF employs column ensemble, i.e., for each input time series, the TSF model will be employed, and their aggregate dictates the final classification label. Table 5 shows the TSF parameters.

$$E = Entopy + \alpha \cdot Margin \quad (14)$$

Table 5. TSF Model Parameters.

Classifier	Type	Input Attributes	Training Set and Test Set
Univariate	Braking Force	Trees: 100	Split—Train: Test = 3:1 (Random Selection)
Multivariate	Braking Force Wheel Slip Motor Temperature Motor Shaft Angular Displacement	Trees: 100	-

4.4. k-Nearest Neighbour Reasoner

The k-Nearest Neighbour (k-NN) method employs instance-based learning, wherein the training usually occurs when a new case is seen. This model is based on memorising cases and their corresponding classes, and upon being introduced to a new case during prediction, the model checks the memory to find a similar case with the closest distance to the new case in the feature space. The value of "k" is user-defined, and it decides the number of closest samples considered for the classification of a new case.

For a classifying time series, Dynamic Time Warping (DTW) needs to be set as the distance metric employed in the k-NN model. DTW is used to measure the similarity between the two-time series. In DTW, points of one-time series are mapped to a corresponding point such that the distance between them is shortest. The k-NN algorithm assigns the test case with the label of the majority class among its "k" number nearest neighbours.

The univariate model intakes the time series attribute braking force, while the multivariate model is fed with the features braking force, wheel slip, motor temperature, and motor shaft angular displacement. For the multivariate model, the features are concatenated into a single feature by the model before employing the DTW. The k-NN parameters are shown in Table 6.

Table 6. k-NN Model Parameters.

Classifier	Type	Input Attributes	Training Set and Test Set
Univariate	Braking Force	Neighbours: 1 Weights: Uniform Metric: DTW	Split—Train: Test = 3:1 (Random Selection)
Multivariate	Braking Force Wheel Slip Motor Temperature Motor Shaft Angular Displacement	Neighbours: 4 Weights: Uniform Metric: DTW	-

5. Results and Discussion

As mentioned previously, each model is evaluated by the criteria of accuracy, precision, recall and F1-score. ML algorithms at large are stochastic or non-deterministic, implying

that the output varies with each run or implementation. Hence, the performance of the model is evaluated in terms of average accuracy, precision, recall and F1-score.

5.1. Univariate Models

Following the reasoners' development, the LSTM model results are shown in Figure 7 and Table 7. It can be seen that the model has wrongly identified two cases of OC (label 1) as jamming faults (label 3) and one instance of jamming as OC. It is also worth noting that all instances of IOC (label 2) were correctly identified, and no false positives were generated for this type of fault. The results obtained for LSTM univariate model are shown in Table 7.

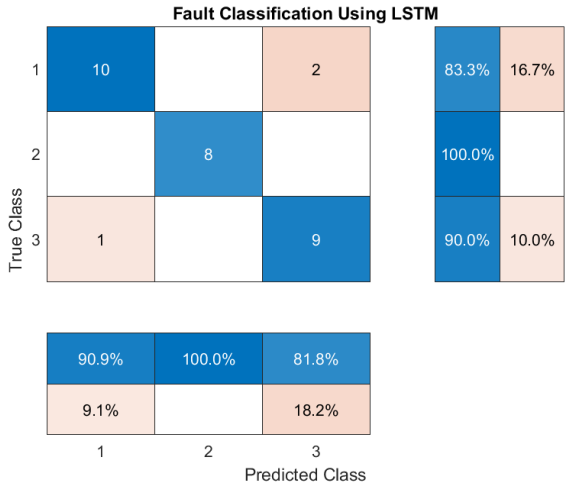

Figure 7. Confusion Matrix for LSTM Univariate Model.

Table 7. LSTM Univariate Performance.

Average Accuracy		**85.3%**	
	Average Precision	**Average Recall**	**Average F1-Score**
OC	89.5%	71.7%	79.4%
IOC	92.8%	100%	96.1%
Jamming	77.1%	90.0%	83.0%

The TSF model showed high accuracy consistently, with the average being 99.34% and not dropping below 97%. The model showcases 100% accuracy for 8 out of 10 iterations. The only misclassification during this iteration is the classification of an instance of IOC as an OC fault. Figure 8 and Table 8 show the TSF confusion matrix and univariate performance values, respectively.

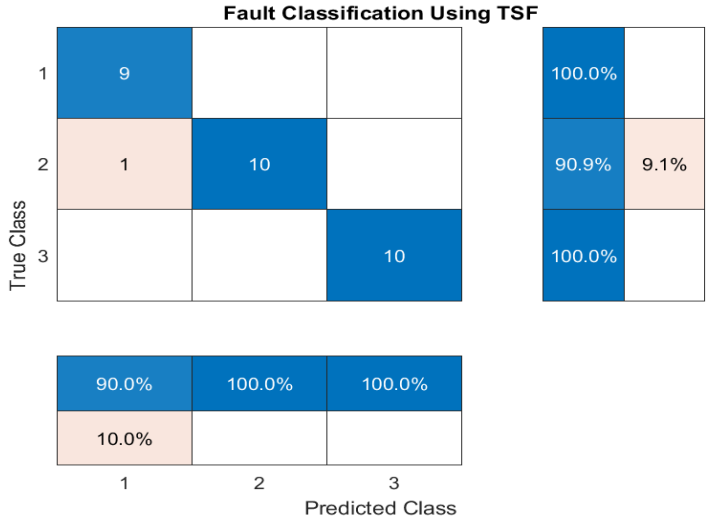

Figure 8. Confusion Matrix for TSF Univariate Model.

Table 8. TSF Univariate Performance.

Average Accuracy		**99.3%**	
	Average Precision	**Average Recall**	**Average F1-Score**
OC	97.9%	100%	98.9%
IOC	100%	98.3%	99.1%
Jamming	100%	100%	100%

Lastly, the k-NN univariate model was tested ten times, and the average accuracy for this model was 72.3%, being the lowest and highest accuracy achieved 80% and 66.7%, respectively. Figure 9 and Table 9 displays the k-NN confusion matrix and performance, respectively. From the confusion matrix, one instance of OC was misclassified as jamming, two IOC as OC, and three cases of jamming as OC. It is also worth noticing that no false negatives were predicted for jamming. It can also be observed that the recall rate for jamming is very low, even though the precision rate is very acceptable.

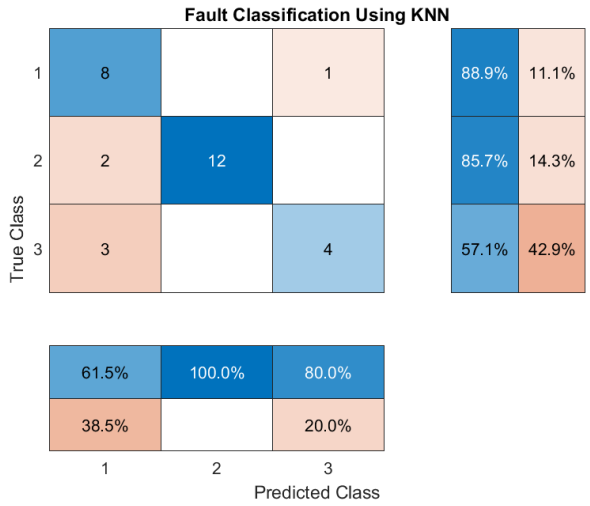

Figure 9. Confusion Matrix for k-NN Univariate Model.

Table 9. k-NN Univariate Performance.

Average Accuracy		**72.3%**	
	Average Precision	**Average Recall**	**Average F1-Score**
OC	62.4%	83.1%	70.8%
IOC	100%	89.8%	94.4%
Jamming	61.6%	42.1%	49.3%

5.2. Multivariate Models

The LSTM multivariate model considers four time series data strings for classifying the input instance. Running ten iterations with the model gave an average accuracy of 30.7%. Figure 10 and Table 10 show the LSTM confusion matrix and performance table, respectively. It can be noted that the model identifies all the test cases as jamming faults. This, in turn, leads to high false-positive cases of jamming faults and simultaneously high false negatives for OC and IOC. From the matrix for the second case, it can be seen that the model labels all of the test instances as IOC, thereby generating a high number of false-positive cases for the same, which also contributes to a high number of false-negative cases for OC and jamming.

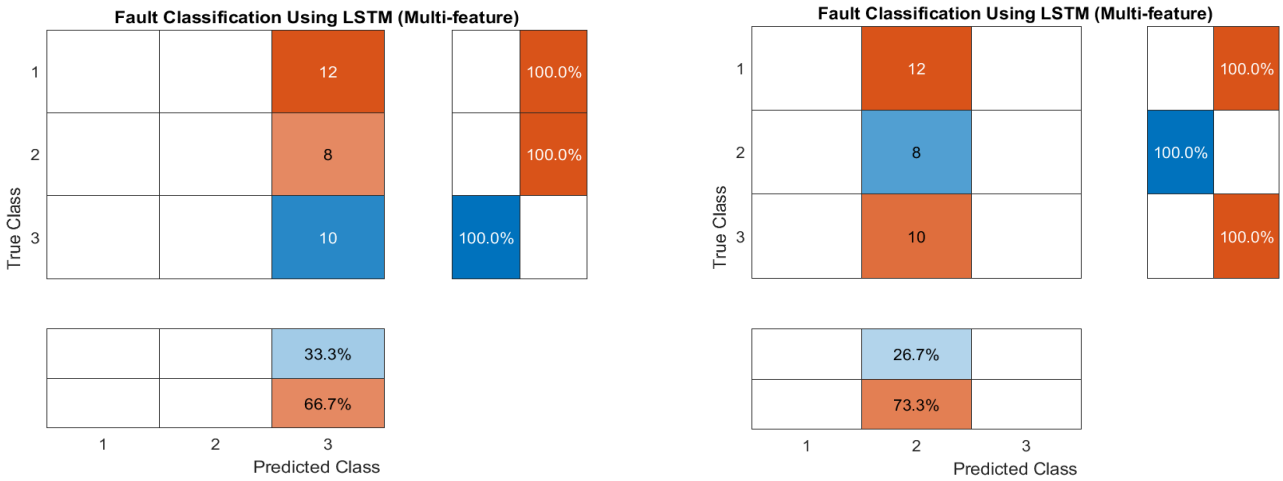

Figure 10. Confusion Matrix for LSTM Model: (**a**) Case 1; (**b**) Case 2.

Table 10. LSTM Multivariate Performance.

Average Accuracy		**30.7%**	
	Average Precision	**Average Recall**	**Average F1-Score**
OC	0%	0%	0%
IOC	10.7%	40.0%	16.9%
Jamming	20.0%	60.0%	30.0%

The performance of the TSF multivariate model is discussed next. The average accuracy of the model is 34.3%, 50% being the highest and 23.3% the lowest. It can be gathered from the confusion matrices that the model is highly susceptible to misclassify OC faults, generating several false-negative alerts for IOC and OC faults. Figure 11 and Table 11 display the model confusion matrix and performance.

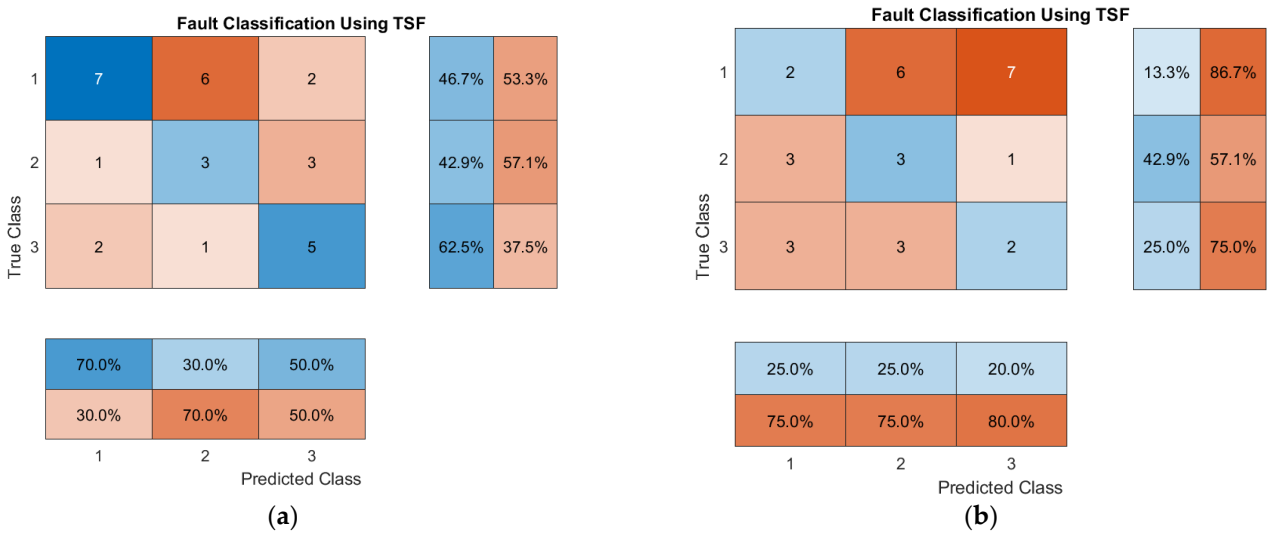

Figure 11. Confusion Matrix for TSF Model: (**a**) Case 1—Most-Accurate Iteration; (**b**) Case 2—Least-Accurate Iteration.

Table 11. TSF Multivariate Performance.

Average Accuracy		**34.3%**	
	Average Precision	**Average Recall**	**Average F1-Score**
OC	47.7%	24.7%	31.9%
IOC	27.0%	35.7%	30.5%
Jamming	31.9%	51.3%	38.7%

The confusion matrix obtained after employing the k-NN in taking multiple features as input is shown below. The overall accuracy of this classifier is 36.7%. The recall, precision and F1-score is 46.7% for OC, which is the most performance shown by this classifier for any type of fault. Figure 12 and Table 12 show the confusion matrix and the model performance, respectively.

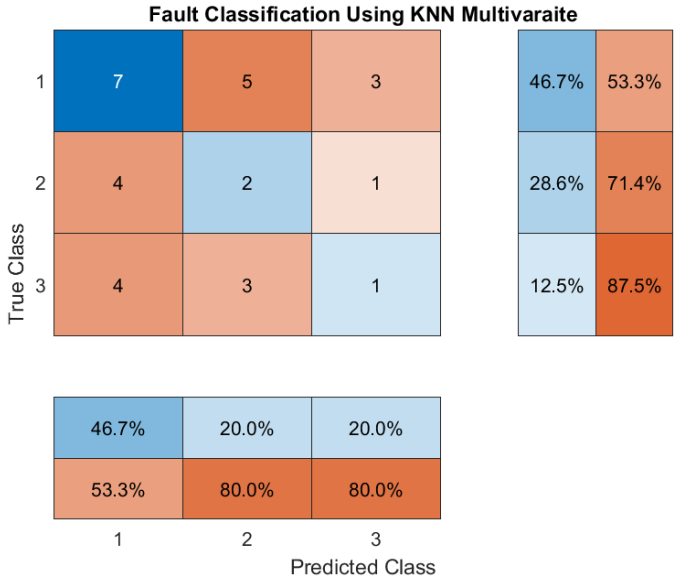

Figure 12. Confusion Matrix for k-NN Multivariate Model.

Table 12. k-NN Multivariate Performance.

Average Accuracy		36.7%	
	Average Precision	**Average Recall**	**Average F1-Score**
OC	46.7%	46.7%	46.7%
IOC	20.0%	28.6%	23.5%
Jamming	20.0%	12.5%	15.4%

5.3. Model Comparison

Accuracy is a good parameter to consider for judging the performance of a system as a whole, but it fails to capture details, such as false-positive predictions or the inability to identify faults that are present. Details such as these are especially important in the context of frequency and criticality of failures that the reasoner is being used to identify.

Here it can be seen that among the univariate models, the reasoner employing the TSF model is the most accurate, with 99.3% accuracy. This is followed by the LSTM model providing 85.3% and, lastly, the k-NN model with 72.3%. Contrary to the univariate models, the k-NN multivariate model is the most accurate of the three models with 36.7% accuracy, followed by the TSF and LSTM with 34.3% and 30.7%, respectively. Accuracy is an effective indicator of performance when the distribution chosen for the dataset for testing is symmetric. For this experiment, the test data are programmed such that it is not always symmetric so as to depict real-life scenarios. Therefore, it will not be appropriate to consider accuracy as a sole indicator of a reasoner performance. Table 13 displays the comparison in model accuracy from the experiment.

Table 13. ML Model Accuracy Comparison.

	Univariate			**Multivariate**		
	LSTM	**TSF**	**k-NN**	**LSTM**	**TSF**	**k-NN**
Accuracy	85.3%	99.3%	72.3%	30.7%	34.3%	36.7%

Another parameter to consider is precision, which in the experiment gives an idea of the ratio of correctly identified OC faults to the total number of OC faults predicted by the model. It can be observed that again, the TSF univariate model provides the highest precision, followed by the LSTM and k-NN models. Among the multivariate models, the LSTM model was unable to identify any faults and the k-NN multivariate was able to achieve a precision of 46.7%. The higher precision of the TSF univariate model is an indicator that it had produced the lowest false positives among the models compared in this experiment. Table 14 show the performance parameters of the OC fault classification.

Table 14. Performance Parameters for OC Classification.

Model	**Average Precision**	**Average Recall**	**Average F1-Score**
LSTM Univariate	89.5%	71.7%	79.4%
TSF Univariate	97.9%	100%	98.9%
k-NN Univariate	62.4%	83.1%	70.8%
LSTM Multivariate	0%	0%	0%
TSF Multivariate	47.7%	24.7%	31.9%
k-NN Multivariate	46.7%	46.7%	46.7%

The recall rate for classifying OC informs the observer of the number of faults that the classifier was able to identify among the total number of OC faults introduced to it. The TSF univariate model has the highest recall rate showcasing the ability to identify all the relevant cases it was shown. The next best value for this metric is showcased by a k-NN univariate model with a recall rate of 83.1%, followed by an LSTM single feature

model with 71.7%, k-NN multivariate with 46.7%, TSF multivariate with 24.7%, and LSTM multi-feature with no recalling ability. It is worth noting that even though the recall rate is good for the k-NN univariate model, the precision rate is around 60%, indicating that it was able to identify a large number of OC faults at the cost of incorrectly classifying some other faults as OC.

F1-score is a measure that gives equal importance to both precision and recall. TSF univariate has the highest score with 98.9%, and the LSTM univariate comes in second with 79.4%. The F1-score for the k-NN univariate model can be said to be a decent 70.8%.

Similarly, for the classification of IOC, both TSF and k-NN univariate models offer 100% precision implying no false-positive cases were recorded. The next best precision is offered by LSTM univariate model with 92.8% precision, followed by TSF, k-NN and LSTM multivariate models with 27.0%, 20.0%, and 10.7%, respectively. Concurrently, the LSTM univariate can identify all the IOC faults it encounters, and the TSF and k-NN single feature models can identify 98.3% and 89.8%, respectively, of all the faults from the set. The multivariate models follow the trend of poor performance it showcased in the OC fault before, with the LSTM model having the ability to recall only 40%, TSF only 35.7%, and k-NN only 28.6% of the relevant instances. The consolidated results for IOC are shown in Table 15.

Table 15. Performance Parameters for IOC Classification.

Model	Average Precision	Average Recall	Average F1-Score
LSTM Univariate	92.8%	100%	96.0%
TSF Univariate	100%	98.3%	99.1%
k-NN Univariate	100%	89.8%	94.4%
LSTM Multivariate	10.7%	40.0%	16.9%
TSF Multivariate	27.0%	35.7%	30.5%
k-NN Multivariate	20.0%	28.6%	23.5%

The TSF univariate model achieves 100% precision, recall and F1-scores for identifying jamming faults, indicating that the predictions had zero false-positive cases while simultaneously successfully managing to identify all the jamming faults it came across. The LSTM single feature provides the next best performance across all three metrics with precision, recall and F1-score of 77.1%, 90% and 83.0%, respectively. The k-NN univariate and the multivariate models have relatively poor performance, with k-NN univariate the best among the rest with a precision of 61.6%, recall of 42.1%, and F1-score of 49.3%. Out of the multivariate models, the LSTM has the highest recalling ability of 60% against the 51.3% provided by TSF and 12.5% provided by k-NN, while TSF offers better precision (31.9%) when compared to LSTM (20%) and k-NN (20%). The k-NN multivariate model offers the least performance with precision, recall and F1-score of 20%, 12.5% and 15.4%, respectively. Table 16. displays the model's performance for the jamming fault classification.

Table 16. Performance Parameters for Jamming Classification.

Model	Average Precision	Average Recall	Average F1-Score
LSTM Univariate	77.1%	90.0%	83.0%
TSF Univariate	100%	100%	100%
k-NN Univariate	61.6%	42.1%	49.3%
LSTM Multivariate	20.0%	60.0%	30.0%
TSF Multivariate	31.9%	51.3%	38.7%
k-NN Multivariate	20.0%	12.5%	15.4%

6. Conclusions

In this project, three faults were simulated, namely, open circuit, intermittent open circuit fault, and jamming faults, in an aircraft's electric braking system. The characteristics of the faults were analysed, and the input features for the models were selected. Following

this, an LSTM model was developed, and its performance was compared against k-NN and TSF univariate and multivariate models.

There are numerous factors to keep in mind while selecting the appropriate model for identifying a fault. For identifying faults that are equally likely to occur, accuracy might be a good indicator, and so employing the univariate TSF machine learning algorithm is recommended because it provides the highest accuracy. In real life, though, the faults rarely occur in symmetric distribution, and so accuracy is not an ideal indicator to measure the performance of the model.

Electrical faults are highly critical, and so it will be more cost-efficient to correctly identify all faults, even if the predictor occasionally generates false positives. Henceforth, it would be ideal to choose models that have high recall metrics, which in this case is the TSF univariate model for OC faults and LSTM univariate for IOC faults. Jamming faults occur less frequently and have a criticality factor less than that of electrical faults, and so precision might be of greater importance. Here, TSF univariate models provide both high precision and recall (100%) in identifying jamming faults, making it the ideal choice.

The decision on what metric is appropriate to evaluate the performance depends on whether the cost incurred due to failure to detect a fault is greater or if the cost associated with grounding an aircraft due to misidentification is more expensive. In the unlikely case that both are equally costly, F1-score is a good metric used to select the model. The TSF univariate model offers the best F1-score for all three faults, making it the obvious choice in this case.

OEMs or airline providers can also choose to either implement multiple algorithms with each dedicated to identifying a specific fault or choose a single model that is selected after ranking the faults depending on their criticality and probability of occurrence and choosing the appropriate metric to select the model.

7. Future Work

A more realistic configuration of the EBS model can be developed in the future, building on the work undertaken in this paper. Applying the methods to a more realistic and wider dataset of rich quality can be carried out to further increase the quality of results as data-driven techniques are heavily dependent on data distribution. In this experiment, data required to train the reasoners were generated from a Simulink model and so represented data from an ideal world. This data can be considered as "clean data", which is orderly and has no noise components. Normally, data obtained from sensors are plagued by noise and often require extensive pre-processing. A more reliable reasoner can be developed if the contribution of this noise is also taken into account as misclassifications commonly occur in classification boundaries, and the addition of noise contributes to this uncertainty. Obtaining actual sensor data from an EBS and training the models will enable us to better predict the performance of the reasoner in live environments and can potentially increase the performance of the existing reasoner as well.

The performance of the ML algorithms deployed by the users can be improved by training it with a larger, richer dataset containing more cases, and also by further fine-tuning the parameters. Furthermore, the reasoner presented in this experiment is developed for offline analysis. An online reasoner for fault detection in EBS could be considered for the future as it would help the maintenance team to be prepared and help them reduce the turn-around time at airports if the faults are minor and easily repairable.

Author Contributions: Conceptualisation, G.R. and S.P.; methodology, G.R. and S.P.; software, G.R.; validation, G.R., S.P., and P.G.; formal analysis, G.R.; investigation, G.R.; resources, G.R. and P.G.; data curation, G.R.; writing—original draft preparation, P.G.; writing—review and editing, P.G. and S.P.; visualisation, G.R., S.P., and P.G.; supervision, S.P.; project administration, G.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Algorithm: Schematic Flowchart for the Reasoner Development.

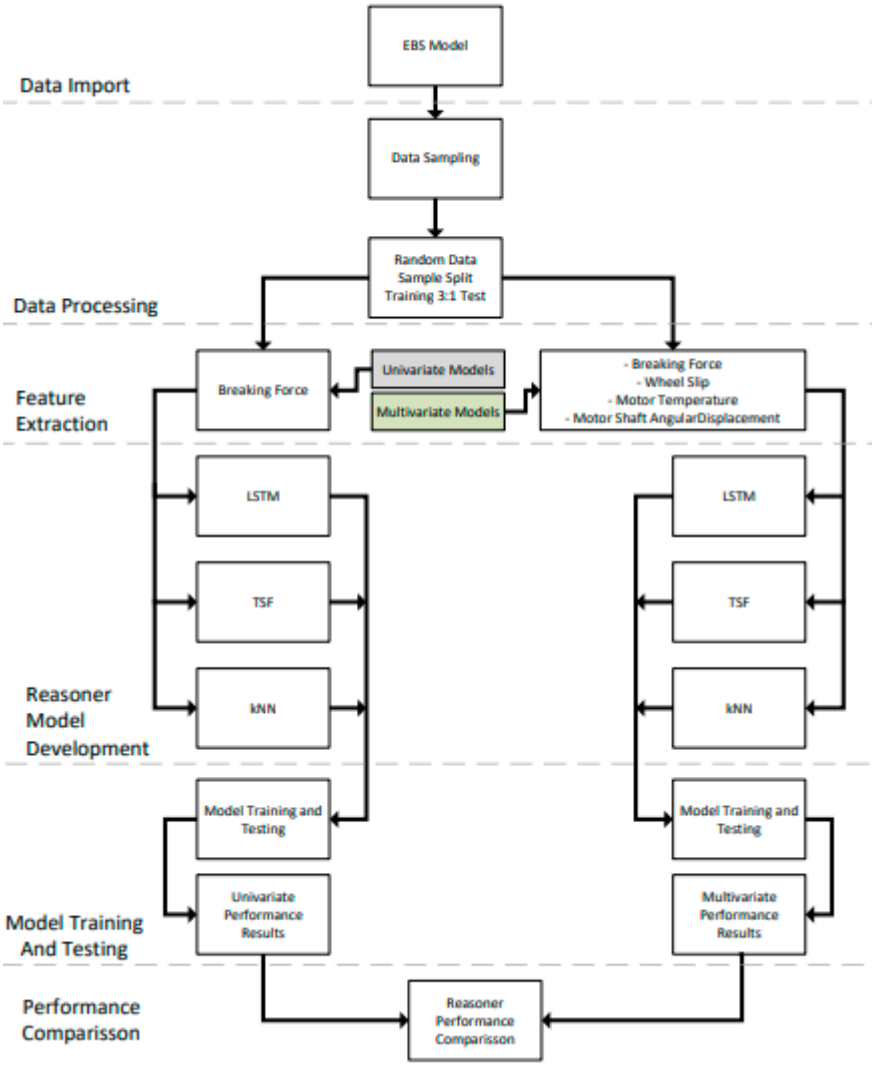

Figure A1. Reasoner Development Flowchart.

References

1. Ezhilarasu, C.M.; Skaf, Z.; Jennions, I.K. The application of reasoning to aerospace Integrated Vehicle Health Management (IVHM): Challenges and opportunities. *Prog. Aerosp. Sci.* **2019**, *105*, 60–73. [CrossRef]
2. Safran. More-Electric Aircraft: To Power the Future. 2020. Available online: https://www.safran-electrical-power.com/media/20150408_more-electric-aircraft-power-future (accessed on 23 May 2021).
3. Aerospace Manufacturing. It's a Twin-Win with Digital! 2020. Available online: https://www.aero-mag.com/ifs-mro-digital-twin-aerospace-15052020/ (accessed on 23 May 2021).
4. Wheeler, P.; Bozhko, S. The More Electric Aircraft: Technology and challenges. *IEEE Electrif. Mag.* **2014**, *2*, 6–12. [CrossRef]
5. Salazar, L.J. Analysing Uncertainty and Delays in Aircraft Heavy Maintenance. Doctoral Thesis, The University of Manchester,, Manchester, UK, 2015.
6. Jennions, I.; Niculita, O.; Esperon-Miguez, M. Integrating IVHM and Asset Design. *Int. J. Progn. Health Manag.* **2016**, *7*. [CrossRef]
7. NASA. Integrated Vehicle Health Management (IVHM). 2008. Available online: https://www.nasa.gov/centers/ames/research/humaninspace/humansinspace-ivhm.html (accessed on 23 May 2021).

8. SAE International. Integrated Vehicle Health Management. 2018. Available online: file:///C:/Users/User/Downloads/IVHM%20Capability%20Matrix%20with%20Definitions.pdf (accessed on 23 May 2021).
9. Park, Y.-J.; Fan, S.-K.S.; Hsu, C.-Y. A Review on Fault Detection and Process Diagnostics in Industrial Processes. *Processes* **2020**, *8*, 1123. [CrossRef]
10. Vohnout, S.; Kim, B.; Kunst, N.; Gleeson, B.; Wagoner, R.; Balaban, E.; Goebel, K. A Model-based Avionic Prognostic Reasoner (MAPR). In Proceedings of the AIAA Infotech at Aerospace Conference and Exhibit, Garden Grove, California, 19–21 June 2012. [CrossRef]
11. Infosys. Aircraft Landing Gear Design & Development. Infosys Ltd., 2018. Available online: https://www.infosys.com/industries/communication-services/documents/landing-gear-design-and-development.pdf (accessed on 19 August 2021).
12. Safran. Braking Systems. 2020. Available online: https://www.safran-landing-systems.com/systems-equipment/braking-system (accessed on 19 August 2021).
13. BAE Systems. Fly-by-Wire Controls Set for Electric Flight. Available online: https://www.baesystems.com/en-us/feature/fly-by-wire-controls-set-for-electric-flight (accessed on 19 August 2021).
14. Huang, H. Challenges in More Electric Aircraft (MEA). *IEEE Transp. Electrif. Community (TEC) eNewsl.* **2015**. Available online: https://tec.ieee.org/newsletter/july-august-2015/challenges-in-more-electric-aircraft-mea (accessed on 19 August 2021).
15. Meyer, H.; Zimdahl, J.; Kamtsiuris, A.; Meissner, R.; Raddatz, F.; Haufe, S.; Bäßler, M. Development of a Digital Twin for Aviation Research. 2020. Available online: https://www.researchgate.net/publication/344046441_DEVELOPMENT_OF_A_DIGITAL_TWIN_FOR_AVIATION_RESEARCH (accessed on 22 May 2021).
16. AIAA. Digital Twin: Definition & Value. 2020. Available online: https://www.aiaa.org/docs/default-source/uploadedfiles/issues-and-advocacy/policy-papers/digital-twin-institute-position-paper-(december-2020).pdf (accessed on 19 August 2021).
17. NASA. Goals and Objectives for Integated Vehicle Health Management (IVHM). Rep. NASA-CR-192656, 1992. Available online: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19930013844.pdf (accessed on 19 August 2021).
18. Van Tung, T.; Yang, B.-S. Machine Fault Diagnosis and Prognosis: The State of The Art. *Int. J. Fluid Mach. Syst.* **2009**, *2*, 61–71. [CrossRef]
19. Garza, P.; Perinpanayagam, S.; Aslam, S.; Wileman, A. Qualitative Validation Approach Using Digital Model for the Health Management of Electromechanical Actuators. *Appl. Sci.* **2020**, *10*, 7809. [CrossRef]
20. Iordanidis, G.; Rees, J. Aircraft Electric Braking System. Available online: https://patents.google.com/patent/EP2878501A1 (accessed on 19 May 2020).
21. Wiedenbrug, E.J. Overheating Electric Motors: A Major Cause of Failure. 2003. Available online: http://www.efficientplantmag.com/2003/04/overheating-electric-motors-a-major-cause-of-failure/ (accessed on 23 May 2021).
22. Han, J.; Kamber, M.; Pei, J. *Data Mining: Concepts and Techniques*, 3rd ed.; Elsevier: Waltham, MA, USA, 2003; ISBN 978-0-12-381479-1.
23. Greg Stanley. Fault Signatures, Pattern Recognition, and Classifiers. 2010. Available online: https://gregstanleyandassociates.com/whitepapers/FaultDiagnosis/Fault-Signatures/fault-signatures.htm (accessed on 23 May 2021).
24. Van Veen, F. The Neural Network Zoo. 2017. Available online: https://www.asimovinstitute.org/author/fjodorvanveen/ (accessed on 23 May 2021).
25. MathWorks (no date ap) Long Short-Term Memory (LSTM). Available online: https://uk.mathworks.com/discovery/lstm.html (accessed on 23 May 2021).
26. MathWorks (no date aq) Long Short-Term Memory Networks. Available online: https://uk.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html (accessed on 23 May 2021).
27. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
28. Weninger, F.; Eyben, F.; Schuller, B. Single-Channel Speech Separation with Memory-Enhanced Recurrent Neural Networks. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 3737–3741.
29. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; Massachusetts Institute of Technology: Cambridge, MA, USA, 2012; ISBN 978-0-262-01802-9.
30. MathWorks (no date ax) ClassificationLayer. Available online: https://www.mathworks.com/help/deeplearning/ref/classificationlayer.html#bu5lho8_sep_bvk87b1 (accessed on 23 May 2021).
31. Deng, H.; Runger, G.; Tuv, E.; Vladimir, M. A time series forest for classification and feature extraction. *Inf. Sci.* **2013**, *239*, 142–153. [CrossRef]

2021-10-02

Digital simulation and identification of faults with neural network reasoners in brushed actuators employed in an e-brake system

Ramesh, Gouri

MDPI

Ramesh G, Garza P, Perinpanayagam S. (2021) Digital simulation and identification of faults with neural network reasoners in brushed actuators employed in an e-brake system. Applied Sciences, Volume 11, Issue 19, Article number 9171

https://doi.org/10.3390/app11199171